

BBBBBBBBBBBBBB AAAAAAAA CCCCCCCCCCCCCC KKK KKK UUU UUU PPPPPPPPPPPPPP
BBBBBBBBBBBBBB AAAAAAAA CCCCCCCCCCCCCC KKK KKK UUU UUU PPPPPPPPPPPPPP
BBBBBBBBBBBBBB AAAAAAAA CCCCCCCCCCCCCC KKK KKK UUU UUU PPPPPPPPPPPPPP
BBB BBB AAA AAA CCC KKK KKK UUU UUU PPP PPP
BBB BBB AAA AAA CCC KKK KKK UUU UUU PPP PPP
BBB BBB AAA AAA CCC KKK KKK UUU UUU PPP PPP
BBB BBB AAA AAA CCC KKK KKK UUU UUU PPP PPP
BBB BBB AAA AAA CCC KKK KKK UUU UUU PPP PPP
BBB BBB AAA AAA CCC KKK KKK UUU UUU PPP PPP
BBB BBB AAA AAA CCC KKK KKK UUU UUU PPP PPP
BBBBBBBBBBBBBB AAA AAA CCC KKKKKKKKK UUU UUU PPPPPPPPPPPPPP
BBBBBBBBBBBBBB AAA AAA CCC KKKKKKKKK UUU UUU PPPPPPPPPPPPPP
BBBBBBBBBBBBBB AAA AAA CCC KKKKKKKKK UUU UUU PPPPPPPPPPPPPP
BBB BBB AAAAAAAA CCC KKK KKK UUU UUU PPP
BBB BBB AAAAAAAA CCC KKK KKK UUU UUU PPP
BBB BBB AAAAAAAA CCC KKK KKK UUU UUU PPP
BBB BBB AAA AAA CCC KKK KKK UUU UUU PPP
BBB BBB AAA AAA CCC KKK KKK UUU UUU PPP
BBB BBB AAA AAA CCC KKK KKK UUU UUU PPP
BBB BBB AAA AAA CCC KKK KKK UUUUUUUUUUUUUUUU PPP
BBBBBBBBBBBBBB AAA AAA CCCCCCCCCCCCCC KKK KKK UUUUUUUUUUUUUUUU PPP
BBBBBBBBBBBBBB AAA AAA CCCCCCCCCCCCCC KKK KKK UUUUUUUUUUUUUUUU PPP
BBBBBBBBBBBBBB AAA AAA CCCCCCCCCCCCCC KKK KKK UUUUUUUUUUUUUUUU PPP

FILE ID**ANALYZE

E 2

```

AAAAAA NN NN AAAAAA LL YY YY ZZZZZZZZ EEEEEEEE
AAAAAA NN NN AAAAAA LL YY YY ZZZZZZZZ EEEEEEEE
AA AA NN NN AA AA LL YY YY ZZ EE
AA AA NN NN AA AA LL YY YY ZZ EE
AA AA NNNN NN AA AA LL YY YY ZZ EE
AA AA NNNN NN AA AA LL YY YY ZZ EE
AA AA NN NN AA AA LL YY YY ZZ EEEE
AA AA NN NN AA AA LL YY YY ZZ EEEE
AAAAAAA NN NNNN AAAAAAAA LL YY ZZ EE
AAAAAAA NN NNNN AAAAAAAA LL YY ZZ EE
AA AA NN NN AA AA LL YY ZZ EE
AA AA NN NN AA AA LL YY ZZ EE
AA AA NN NN AA AA LLLLLLLL YY ZZZZZZZZ EEEE
AA AA NN NN AA AA LLLLLLLL YY ZZZZZZZZ EEEE

```

```
1 0001 0 MODULE ANALYZE (%TITLE 'Analyze a save set'
2 0002 0 IDENT = 'V04-000'
3 0003 0 )
4 0004 1 BEGIN
5 0005 1
6 0006 1
7 0007 1 ****
8 0008 1 *
9 0009 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
10 0010 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
11 0011 1 * ALL RIGHTS RESERVED.
12 0012 1 *
13 0013 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
14 0014 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
15 0015 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
16 0016 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
17 0017 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
18 0018 1 * TRANSFERRED.
19 0019 1 *
20 0020 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
21 0021 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
22 0022 1 * CORPORATION.
23 0023 1 *
24 0024 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
25 0025 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
26 0026 1 *
27 0027 1 *
28 0028 1 ****
29 0029 1
30 0030 1
31 0031 1 ++
32 0032 1 FACILITY:
33 0033 1 Backup/Restore
34 0034 1
35 0035 1 ABSTRACT:
36 0036 1 This module contains the routines that analyze a save set.
37 0037 1
38 0038 1 ENVIRONMENT:
39 0039 1 VAX/VMS user mode.
40 0040 1 --
41 0041 1
42 0042 1 AUTHOR: M. Jack, CREATION DATE: 03-Sep-1980
43 0043 1
44 0044 1 MODIFIED BY:
45 0045 1
46 0046 1 V03-009 LY0510 Larry Yetto 19-JUL-1984 08:44
47 0047 1 Add support for the new longword devtyp in the physical
48 0048 1 volume attributes record. The format of this longword is
49 0049 1 the same as UCB$L_MEDIA_ID
50 0050 1
51 0051 1 V03-008 LY0485 Larry Yetto 27-APR-1984 08:42
52 0052 1 FT1 QAR # 2088 - If the saveset being read is encrypted
53 0053 1 and /ENCRYPT not specified then report an error
54 0054 1
55 0055 1 V03-007 LMP0176 L. Mark Pilant, 6-Dec-1983 10:41
56 0056 1 Use the correct width in the call to $FORMAT_ACL.
57 0057 1
```

58 0058 1 | V03-006 JWT0137 Jim Teague 19-Sep-1983 08:22
59 0059 1 | CRYPTO_INIDEC and CRYPTO_DECR_BLOCK need to be WEAKened.
60 0060 1 |
61 0061 1 | V03-005 JEP0003 J. Eric Pollack, 23-Apr-1983 10:53
62 0062 1 | Add support for encrypted savesets.
63 0063 1 |
64 0064 1 | V03-004 ACG0332 Andrew C. Goldstein, 19-Apr-1983 18:10
65 0065 1 | Add support for file highwater mark and RMS journal flags
66 0066 1 |
67 0067 1 | V03-003 LMP0100 L. Mark Pilant, 14-Apr-1983 13:17
68 0068 1 | Add te \$FORMAT_ACL system service.
69 0069 1 |
70 0070 1 | V03-002 ACG0313 Andrew C. Goldstein, 12-Feb-1983 16:01
71 0071 1 | Add routine subtitles
72 0072 1 |
73 0073 1 | V03-001 LMP0044 L. Mark Pilant, 21-Oct-1982 15:10
74 0074 1 | Add support for ACL's.
75 0075 1 |
76 0076 1 | V02-006 MLJ0081 Martin L. Jack, 26-Feb-1982 16:16
77 0077 1 | Add RETAINMIN and RETAINMAX attributes to support new home
78 0078 1 | block fields.
79 0079 1 |
80 0080 1 | V02-005 MLJ0075 Martin L. Jack, 28-Jan-1982 20:02
81 0081 1 | Add VERLIMIT and DIR_VERLIM attributes to support version limit
82 0082 1 | handling.
83 0083 1 |
84 0084 1 | V02-004 MLJ0062 Martin L. Jack, 3-Dec-1981 12:16
85 0085 1 | Add DIR_STATUS attribute to support /INCREMENTAL.
86 0086 1 |
87 0087 1 | V02-003 MLJ0036 Martin L. Jack, 28-Aug-1981 17:09
88 0088 1 | Implement parent directory attributes and reel restart.
89 0089 1 |
90 0090 1 | V02-002 MLJ0023 Martin L. Jack, 23-Apr-1981 11:36
91 0091 1 | Implement placement attribute.
92 0092 1 |
93 0093 1 | V02-001 MLJ0010 Martin L. Jack, 25-Mar-1981 14:58
94 0094 1 | Add new attributes for image restore. Make some routines
95 0095 1 | common with LIST module. Replace OWN storage with LOCAL.
96 0096 1 | Change !SL directives to !UL.
97 0097 1 |
98 0098 1 | **

```
100 0099 1 REQUIRE 'SRC$:COMMON';
101 1205 1 LIBRARY 'SYSSLIBRARY:$STARLET';
102 1206 1 REQUIRE 'LIB$:BACKDEF';
103 1656 1
104 1657 1
105 1658 1 LINKAGE
106 1659 1 L_PS = CALL: GLOBAL(P$=11);
107 1660 1
108 1661 1
109 1662 1 MACRO
110 1663 1 L_DECL = EXTERNAL REGISTER PS = 11: REF VECTOR %;
111 1664 1
112 1665 1
113 1666 1 FORWARD ROUTINE
114 1667 1 ANALYZE_ONE_ATTRIBUTE:
115 1668 1 L_PS NOVALUE, ! Format an attribute record
116 1669 1 ANALYZE_ONE_RECORD:
117 1670 1 L_PS NOVALUE, ! Format a record
118 1671 1 ANALYZE_ONE_BUFFER:
119 1672 1 L_PS NOVALUE, ! Format a block
120 1673 1 ANALYZE: NOVALUE; ! Driver for save set analysis
121 1674 1
122 1675 1
123 1676 1 EXTERNAL ROUTINE
124 1677 1 DEBLOCK: L_PS NOVALUE, ! Deblock a save set buffer
125 1678 1 DEBLOCK ATTR: L_PS NOVALUE, ! Deblock an attribute record
126 1679 1 DECODE_DEVTYPE: NOVALUE, ! Convert DEVTYPE to ASCII string
127 1680 1 FIN_IN_SAVE: NOVALUE, ! Finish input save set processing
128 1681 1 INIT_IN_SAVE: NOVALUE, ! Initialize input save set processing
129 1682 1 LIST_FA0: L_PS NOVALUE, ! Add information to line buffer
130 1683 1 LIST_EOL: L_PS NOVALUE, ! Write line buffer to listing file
131 1684 1 LIST_PROTECTION:L_PS NOVALUE, ! List protection code
132 1685 1 READ_BUFFER, ! Get a save set buffer
133 1686 1 RESTORE_HANDLER, ! Handler for RESTORE, LIST, ANALYZE
134 1687 1 CRYPTO_INIDEC: WEAK, ! Initialize for reading encrypted saveset
135 1688 1 CRYPTO_CHKSAV, ! Check if saveset is encrypted
136 1689 1 CRYPTO_DECR_BLOCK: NOVALUE
137 1690 1 WEAK: ! Decrypt one block
138 1691 1
139 1692 1
140 1693 1 GSDEFINE(); ! Define global common area
141 1694 1
142 1695 1
143 1696 1 BIND
144 1697 1 FALSETRUE = UPLIT(
145 1698 1 UPLIT BYTE (%ASCII 'False'),
146 1699 1 UPLIT BYTE (%ASCII 'True'))
147 1700 1 : VECTOR;
148 1701 1
149 1702 1
150 1703 1 EXTERNAL LITERAL
151 1704 1 BACKUPS_BACNOTENC,
152 1705 1 BACKUPS_ENCSAVSET;
153 1706 1
154 1707 1 MACRO
155 1708 1 FAO_(A)=
156 M 1709 1 LIST_FA0(
```

```
157 M 1710 1 UPLIT BYTE (%ASCIC A)
158 1711 1 %IF NOT %NULL(%REMAINING) %THEN , %FI %REMAINING) %,
159 1712 1
160 1713 1
161 M 1714 1 EOL_(A)=
162 1715 1 LIST_EOL() %;
163 1716 1
164 1717 1
165 1718 1 MACRO
166 1719 1 _LIST_DESC= PS[0] %, Descriptor for buffer
167 1720 1 _LIST_BUFFER= PS[2] %, Listing buffer
168 1721 1 [IST_DESC(N)= VECTOR[_LIST_DESC,N] %;
169 1722 1
170 1723 1
171 1724 1 LITERAL
172 1725 1 PS$SIZE= 2 + CH$ALLOCATION(LIST_SIZE);
```

```
174 1726 1 %SBTTL 'ANALYZE_ONE_ATTRIBUTE - analyze contents of attribute record'
175 1727 1 ROUTINE ANALYZE_ONE_ATTRIBUTE(ATT): [_PS NOVALUE=
176 1728 1
177 1729 1 !++
178 1730 1
179 1731 1 FUNCTIONAL DESCRIPTION:
180 1732 1 This routine analyzes the contents of one attribute record.
181 1733 1
182 1734 1 INPUT PARAMETERS:
183 1735 1 ATT - Pointer to attribute record.
184 1736 1
185 1737 1 IMPLICIT INPUTS:
186 1738 1 NONE
187 1739 1
188 1740 1 OUTPUT PARAMETERS:
189 1741 1 NONE
190 1742 1
191 1743 1 IMPLICIT OUTPUTS:
192 1744 1 NONE
193 1745 1
194 1746 1 ROUTINE VALUE:
195 1747 1 NONE
196 1748 1
197 1749 1 SIDE EFFECTS:
198 1750 1 The listing is produced.
199 1751 1
200 1752 1 !--
201 1753 1
202 1754 2 BEGIN
203 1755 2
204 1756 2 LITERAL
205 1757 2 DEVTYP_BUF_LEN = 5 ;
206 1758 2
207 1759 2 MAP
208 1760 2 ATT: REF BBLOCK; ! Pointer to attribute record
209 1761 2
210 1762 2 BIND
211 1763 2 ATTRS = UPLIT(
212 1764 2     UPLIT BYTE (%ASCIC 'SSNAME'),
213 1765 2     UPLIT BYTE (%ASCIC 'COMMAND'),
214 1766 2     UPLIT BYTE (%ASCIC 'COMMENT'),
215 1767 2     UPLIT BYTE (%ASCIC 'USERNAME'),
216 1768 2     UPLIT BYTE (%ASCIC 'USERUIC'),
217 1769 2     UPLIT BYTE (%ASCIC 'DATE'),
218 1770 2     UPLIT BYTE (%ASCIC 'OPSYS'),
219 1771 2     UPLIT BYTE (%ASCIC 'SYSVER'),
220 1772 2     UPLIT BYTE (%ASCIC 'NODENAME'),
221 1773 2     UPLIT BYTE (%ASCIC 'SIR'),
222 1774 2     UPLIT BYTE (%ASCIC 'DPIVEID'),
223 1775 2     UPLIT BYTE (%ASCIC 'BACKVER'),
224 1776 2     UPLIT BYTE (%ASCIC 'BLOCKSIZE'),
225 1777 2     UPLIT BYTE (%ASCIC 'XORSIZE'),
226 1778 2     UPLIT BYTE (%ASCIC 'BUFFERS'),
227 1779 2     UPLIT BYTE (%ASCIC 'VOLSETNAM'),
228 1780 2     UPLIT BYTE (%ASCIC 'NVOLS'),
229 1781 2     UPLIT BYTE (%ASCIC 'BACKSIZE'),
230 1782 2     UPLIT BYTE (%ASCIC 'BACKFILES'),
```

231 1783 2 UPLIT BYTE (%ASCIC 'VOLSTRUCT').
232 1784 2 UPLIT BYTE (%ASCIC 'VOLNAME').
233 1785 2 UPLIT BYTE (%ASCIC 'OWNERNAME').
234 1786 2 UPLIT BYTE (%ASCIC 'FORMAT').
235 1787 2 UPLIT BYTE (%ASCIC 'RVN').
236 1788 2 UPLIT BYTE (%ASCIC 'VOLOWNER').
237 1789 2 UPLIT BYTE (%ASCIC 'PROTECT').
238 1790 2 UPLIT BYTE (%ASCIC 'FILEPROT').
239 1791 2 UPLIT BYTE (%ASCIC 'RECPROT').
240 1792 2 UPLIT BYTE (%ASCIC 'VOLCHAR').
241 1793 2 UPLIT BYTE (%ASCIC 'VOLDATE').
242 1794 2 UPLIT BYTE (%ASCIC 'WINDOW').
243 1795 2 UPLIT BYTE (%ASCIC 'LRU LIM').
244 1796 2 UPLIT BYTE (%ASCIC 'EXTEND').
245 1797 2 UPLIT BYTE (%ASCIC 'CLUSTER').
246 1798 2 UPLIT BYTE (%ASCIC 'RESFILES').
247 1799 2 UPLIT BYTE (%ASCIC 'VOLSIZE').
248 1800 2 UPLIT BYTE (%ASCIC 'TOTSIZE').
249 1801 2 UPLIT BYTE (%ASCIC 'TOTFILES').
250 1802 2 UPLIT BYTE (%ASCIC 'MAXFILES').
251 1803 2 UPLIT BYTE (%ASCIC 'MAXFILNUM').
252 1804 2 UPLIT BYTE (%ASCIC 'SERIALNUM').
253 1805 2 UPLIT BYTE (%ASCIC 'FILENAME').
254 1806 2 UPLIT BYTE (%ASCIC 'STRUCLEV').
255 1807 2 UPLIT BYTE (%ASCIC 'FID').
256 1808 2 UPLIT BYTE (%ASCIC 'BACKLINK').
257 1809 2 UPLIT BYTE (%ASCIC 'FILESIZE').
258 1810 2 UPLIT BYTE (%ASCIC 'UIC').
259 1811 2 UPLIT BYTE (%ASCIC 'FPRO').
260 1812 2 UPLIT BYTE (%ASCIC 'RPRO').
261 1813 2 UPLIT BYTE (%ASCIC 'ACLEVEL').
262 1814 2 UPLIT BYTE (%ASCIC 'UCHAR').
263 1815 2 UPLIT BYTE (%ASCIC 'RECATTR').
264 1816 2 UPLIT BYTE (%ASCIC 'REVISION').
265 1817 2 UPLIT BYTE (%ASCIC 'CREDATE').
266 1818 2 UPLIT BYTE (%ASCIC 'REVDATE').
267 1819 2 UPLIT BYTE (%ASCIC 'EXPDATE').
268 1820 2 UPLIT BYTE (%ASCIC 'BAKDATE').
269 1821 2 UPLIT BYTE (%ASCIC 'SECTORS').
270 1822 2 UPLIT BYTE (%ASCIC 'TRACKS').
271 1823 2 UPLIT BYTE (%ASCIC 'CYLINDERS').
272 1824 2 UPLIT BYTE (%ASCIC 'MAXBLOCK').
273 1825 2 UPLIT BYTE (%ASCIC 'DEVTYP').
274 1826 2 UPLIT BYTE (%ASCIC 'SERIAL').
275 1827 2 UPLIT BYTE (%ASCIC 'DEVNAM').
276 1828 2 UPLIT BYTE (%ASCIC 'LABEL').
277 1829 2 UPLIT BYTE (%ASCIC 'BADBLOCK').
278 1830 2 UPLIT BYTE (%ASCIC 'INDEXLBN').
279 1831 2 UPLIT BYTE (%ASCIC 'BOOTBLOCK').
280 1832 2 UPLIT BYTE (%ASCIC 'BOOTVBN').
281 1833 2 UPLIT BYTE (%ASCIC 'PLACEMENT').
282 1834 2 UPLIT BYTE (%ASCIC 'DIR UIC').
283 1835 2 UPLIT BYTE (%ASCIC 'DIR-FPRO').
284 1836 2 UPLIT BYTE (%ASCIC 'DIR-STATUS').
285 1837 2 UPLIT BYTE (%ASCIC 'DIR-VERLIM').
286 1838 2 UPLIT BYTE (%ASCIC 'VERLIMIT').
287 1839 2 UPLIT BYTE (%ASCIC 'RETAINMIN').

```
288 1840 2 UPLIT BYTE (%ASCIC 'RETAINMAX'),  
289 1841 2 UPLIT BYTE (%ASCIC 'ACLSEGMENT'),  
290 1842 2 UPLIT BYTE (%ASCIC 'HIGHWATER'),  
291 1843 2 UPLIT BYTE (%ASCIC 'JNL FLAGS'),  
292 1844 2 UPLIT BYTE (%ASCIC 'CRYPTDATKEY'))  
293 1845 2 : VECTOR;  
294 1846 2 L_DECL;  
295 1847 2  
296 1848 2  
297 1849 2 ! List the attribute name.  
298 1850 2  
299 P 1851 2 FAO_('      SIZE = !3SL, TYPE = !AC'  
300 1852 2 .ATT[BSASW_SIZE], .ATTRSC.ATT[BSASW_TYPE]-1));  
301 1853 2 EOL_();  
302 1854 2  
303 1855 2  
304 1856 2 ! List the attribute value in an appropriate format.  
305 1857 2  
306 1858 2 FAO('      '):  
307 1859 2 CASE .ATT[BSASW_TYPE] FROM BSASK_SSNAME TO BSASK_NUM_ATRS-1 OF  
308 1860 2 SET  
309 1861 2  
310 1862 2 [BSASK_SSNAME, BSASK_COMMAND, BSASK_COMMENT, BSASK_USERNAME, BSASK_SYSVER,  
311 1863 2 BSASK_BACKVER, BSASR_NODENAME, BSASK_DRIVEID, BSASK_VOLSETNAM,  
312 1864 2 BSASK_VOLNAME, BSASK_OWNERNAME, BSASR_FORMAT, BSASK_FILENAME,  
313 1865 2 BSASK_DEVNAM, BSASK_LABEL]:  
314 P 1866 2 FAO_('!!!AF'  
315 1867 2 .ATT[BSASW_SIZE], ATT[BSASC_LENGTH,0,0,0]);  
316 1868 2  
317 P 1869 2 [BSASK_USERUIC, BSASK_VOLOWNER, BSASK_UIC, BSASK_DIR_UIC]:  
318 1870 2 FAO_('!%U'  
319 1871 2 .ATT[BSASC_LENGTH,0,32,0]);  
320 1872 2  
321 1873 2 [BSASK_DATE, BSASK_VOLDATE, BSASK_CREDATE, BSASK_REVDATE, BSASK_EXPDATE,  
322 1874 2 BSASK_BAKDATE, BSASK_RETAINMIN, BSASK_RETAINMAX]:  
323 P 1875 2 FAO_('!%D'  
324 1876 2 .ATT[BSASC_LENGTH,0,0,0]);  
325 1877 2  
326 1878 2 [BSASK_BLOCKSIZE, BSASK_XORSIZE, BSASK_BUFFERS, BSASK_NVOLS,  
327 1879 2 BSASK_BACKFILES, BSASK_RVN, BSASK_WINDOW, BSASK_LRU [IM, BSASK_EXTEND,  
328 1880 2 BSASK_CLUSTER, BSASK_RESFILES, BSASK_VOLSIZE, BSASK_TOTFILES,  
329 1881 2 BSASK_MAXFILES, BSASR_MAXFILNUM, BSASK_FILESIZ, BSASK_REVISION,  
330 1882 2 BSASK_SECTORS, BSASK_TRACKS, BSASK_CYLINDERS, BSASK_MAXBLOCK,  
331 1883 2 BSASK_INDEXLBN, BSASR_BOOTVBN, BSASK_DIR_VERIM,  
332 1884 2 BSASK_VERLIMIT, BSASK_HIGHWATER]:  
333 P 1885 2 FAO_('!UL'  
334 1886 2 .ATT[BSASC_LENGTH,0,.ATT[BSASW_SIZE]*8,1]);  
335 1887 2  
336 1888 2 [BSASK_DEVTYP]:  
337 1889 2 IF .ATT[BSASW_SIZE] EQ 1  
338 1890 2 THEN  
339 1891 2 ! Old format DEVTYPE attribute. This is the DEVTYPE from the UCB  
340 1892 2  
341 P 1893 2 FAO_('!UL'  
342 1894 2 .ATT[BSASC_LENGTH,0,.ATT[BSASW_SIZE]*8,1])  
343 1895 2  
344 1896 2 ELSE
```

```

345 1897 3      BEGIN
346 1898 3
347 1899 3      ! New format DEVTYP.  We now use a longword and store
348 1900 3      the MEDIA_ID from the UCB.  We use the nondecoded
349 1901 3      form of the MEDIA_ID so we must now pull the ASCII out.
350 1902 3
351 1903 3      LOCAL
352 1904 3      NAME_LENGTH : LONG INITIAL (DEVTYP_BUF_LEN),
353 1905 3      TYPE_LENGTH : LONG INITIAL (DEVTYP_BUF_LEN),
354 1906 3      NAME_BUFFER : VECTOR[DEVTYP_BUF_LEN,BYTE]
355 1907 3      TYPE_BUFFER : VECTOR[DEVTYP_BUF_LEN,BYTE] ;
356 1908 3
357 1909 3      DECODE_DEVTYPE ( .ATT[BSASC_LENGTH,0,32,0],
358 1910 3      NAME_LENGTH, NAME_BUFFER,
359 1911 3      TYPE_LENGTH, TYPE_BUFFER ) ;
360 1912 3
361 P 1913 3      FAO_ ('!XL (!AF,!AF)'
362 P 1914 3      .ATT[BSASC_LENGTH,0,.ATT[BSASW_SIZE]*8,0],
363 P 1915 3      .TYPE_LENGTH, TYPE_BUFFER,
364 P 1916 3      .NAME_LENGTH, NAME_BUFFER' );
365 1917 3
366 1918 2
367 1919 2
368 1920 2
369 1921 2      [BSASK_FID, BSASK_BACKLINK]:
370 P 1922 2      FAO_ ('!UL,!UL,!UL'
371 P 1923 2      .ATT[BSASC_LENGTH,0,16,0] + .ATT[BSASC_LENGTH+5,0,8,0] ^ 16,
372 P 1924 2      .ATT[BSASC_LENGTH+2,0,16,0],
373 P 1925 2      .ATT[BSASC_LENGTH+4,0,8,0]);
374 1926 2
375 1927 2
376 1928 2      [BSASK_PROTECT]:
377 1929 2      LIST_PROTECTION(.ATT[BSASC_LENGTH,0,16,0], 'RWCD');
378 1930 2
379 1931 2      [BSASK_FILEPROT, BSASK_FPRO, BSASK_DIR_FPRO]:
380 1932 2      LIST_PROTECTION(.ATT[BSASC_LENGTH,0,16,0], 'RWED');
381 1933 2
382 1934 2      [BSASK_RECPROT, BSASK_RPRO]:
383 1935 2      LIST_PROTECTION(.ATT[BSASC_LENGTH,0,16,0], 'RWUD');
384 1936 2
385 1937 3      [BSASK_BADBLOCK]:
386 1938 3      BEGIN LOCAL P: REF VECTOR;
387 1939 3      P = ATT[BSASC_LENGTH,0,0,0];
388 1940 4      WHILE .P LSSA ATT[BSASC_LENGTH,0,0,0]+ATT[BSASW_SIZE] DO
389 1941 4      BEGIN
390 1942 4      FAO_ ('!UL:!UL ', .P[0], .P[1]);
391 1943 3      P = .P + 8;
392 1944 2      END;
393 1945 2
394 1946 2      [BSASK_SERIAL, BSASK_SERIALNUM]:
395 P 1947 2      FAO_ ('!OL'
396 1948 2      .ATT[BSASC_LENGTH,0,32,0]);
397 1949 2
398 1950 2      [BSASK_ACLSEGMENT]:
399 1951 3      BEGIN
400 1952 3      LOCAL
401 1953 3      ACE_POINTER : REF BBLOCK.           ! Address of the current ACE

```

```
402      1954 3      ACE_BINDESC : BBLOCK [8],          ! ACE binary descriptor
403      1955 3      ACE_TXTDESC : BBLOCK [8],          ! ACE text descriptor
404      1956 3      ACE_TEXT : BBLOCK [512];          Converted ACE storage
405      1957 3      ACE_POINTER = ATT[BSASC_LENGTH,0,0,0];  Start of ACE's
406      1958 3      CHSFILL (0, 8, ACE_BINDESC);
407      1959 3      CHSFILL (0, 8, ACE_TXTDESC);
408      1960 3      UNTIL .ACE_POINTER GEQA ATT[BSASC_LENGTH,0,0,0] + .ATT[BSASW_SIZE]
409      1961 3      DO
410      1962 4      BEGIN
411      1963 4      ACE_BINDESC[DSCSW_LENGTH] = .ACE_POINTER[ACESB_SIZE];
412      1964 4      ACE_BINDESC[DSCSA_POINTER] = .ACE_POINTER;
413      1965 4      ACE_TXTDESC[DSCSW_LENGTH] = 512;
414      1966 4      ACE_TXTDESC[DSCSA_POINTER] = ACE_TEXT;
415      P 1967 4      $FORMAT_ACL (ACLEN = ACE_BINDESC,
416      P 1968 4          ACLEN = ACE_TXTDESC[DSCSW_LENGTH],
417      P 1969 4          ACLSTR = ACE_TXTDESC,
418      P 1970 4          WIDTH = %REF (80),
419      P 1971 4          TRMDSC = $DESCRIPTOR (%CHAR (13), %CHAR (10)),
420      1972 4          INDENT = %REF (6));
421      1973 4
422      1974 4
423      1975 4      ! Shave off the first 6 blanks if this is the first ACE being
424      1976 4      output as they have already been output.
425      1977 4
426      1978 4      IF .ACE_POINTER EQLA ATT[BSASC_LENGTH,0,0,0]
427      1979 4      THEN
428      1980 5      BEGIN
429      1981 5          ACE_TXTDESC[DSCSW_LENGTH] = .ACE_TXTDESC[DSCSW_LENGTH] - 6;
430      1982 5          ACE_TXTDESC[DSCSA_POINTER] = .ACE_TXTDESC[DSCSA_POINTER] + 6;
431      1983 4      END;
432      1984 4      FAO_ ('!AS', ACE_TXTDESC);
433      1985 4      EOL_ ();
434      1986 4      ACE_POINTER = .ACE_POINTER + .ACE_POINTER[ACESB_SIZE];
435      1987 3
436      1988 2      END;
437      1989 2
438      1990 2      [BSASK CRYPTKEY] :
439      1991 3      BEGIN
440      1992 3          FAO_ ('Encrypted with algorithm: !XB', .ATT[BSASB_CRYPTYP]);
441      1993 3          EOL_ ();
442      1994 3          IF .ATT[BSASB_CRYPTYP] NEQU 0
443      1995 3          THEN
444      1996 4          BEGIN
445      P 1997 4              FAO_ ('Key:!XL !XL, Iv:!XL !XL',
446      P 1998 4                  .ATT[$BYTEOFFSET(BSASQ_CRYPTKEY),0,32,0]
447      P 1999 4                  .ATT[$BYTEOFFSET(BSASQ_CRYPTKEY)+4,0,32,0],
448      P 2000 4                  .ATT[$BYTEOFFSET(BSASQ_CRYPTIV),0,32,0]
449      P 2001 4                  .ATT[$BYTEOFFSET(BSASQ_CRYPTIV)+4,0,32,0]);
450      2002 4          EOL_ ();
451      2003 3
452      2004 2
453      2005 2
454      2006 2      [INRANGE, OUTRANGE]:
455      2007 3      BEGIN
456      2008 3          DEC I FROM .ATT[BSASW_SIZE]-1 TO 0 DO
457      2009 3              FAO_ ('!XB', .ATT[I+BSASC_LENGTH,0,8,0]);
458      2010 2          END;
```

: 459
: 460
: 461
: 462 2011 2 2012 2 TES;
2013 2 EOL();
2014 1 END;

.TITLE ANALYZE Analyze a save set
.IDENT \V04-000\
.PSECT COMMON,NOEXE, GVR,2
00000 GLOBAL_BASE:
00000 .BLKB 0
00000 FREE_LIST:
00008 .BLKB 8
00008 INPUT_WAIT:
00010 .BLKB 8
00010 REREAD_WAIT:
00018 .BLKB 8
00018 OUTPUT_WAIT:
00020 .BLKB 8
00020 JPI_UIC:.BLKB 4
00024 JPI_USERNAME:
00024 .BLKB 12
00030 JPI_DATE:
00038 .BLKB 8
00038 JPI_NODE_DESC:
00040 .BLKB 8
00040 JPI_CURPRIV:
00048 .BLKB 8
00048 SYI_VERSION:
0004C .BLKB 4
0004C SYI_SID:.BLKB 4
00050 RWSV_HOLD_LIST:
00050 .BLKB 8
00058 RWSV_CRC16:
00058 .BLKB 64
00098 RWSV_AUTODIN:
00098 .BLKB 64
000D8 RWSV_FILESET_ID:
000D8 .BLKB 8
000E0 RWSV_VOLUME_ID:
000E0 .BLKB 12
000EC RWSV_VOL_NUMBER:
000EC .BLKB 2
000EE RWSV_SEG_NUMBER:
000EE .BLKB 2
000F0 RWSV_FILE_NUMBER:
000F0 .BLKB 4
000F4 RWSV_SAVE_QUAL:
000F4 .BLKB 4
000F8 RWSV_SAVE_FAB:
000F8 .BLKB 4
000FC RWSV_CHAN:
000FC .BLKB 4
00100 RWSV_XOR_BCB:
00100 .BLKB 4

00104 RWSV_IN_SEQ:
00108 RWSV_IN_SEQ 0:
0010C RWSV_IN_XOR SEG:
00110 RWSV_IN_XOR RFA:
00116 RWSV_LOOKAHEAD:
00117 RWSV_XORSIZE:
00118 RWSV_IN_GROUP_SIZE:
0011C RWSV_IN_ERRORS:
0011E RWSV_IN_XORUSE:
00120 RWSV_IN_ORGERR:
00128 RWSV_IN_VBN:
0012C RWSV_IN_VBN 0:
00130 RWSV_ALLOC:
00134 RWSV_EOF:
00138 RWSV_OUT_SEQ:
0013C RWSV_OUT_VBN:
00140 RWSV_OUT_BLOCK_COUNT:
00144 RWSV_OUT_ERRORS:
00146 RWSV_SEQ_ERRORS:
00148 RWSV_OUT_GROUP_COUNT:
00149 RWSV_PADDING:
0014C QUAL: .BLKB 112
001BC COM_SSNAME:
001C4 COM_VALID_TYPES:
001C6 COM_FLAGS:
001C8 COM_PADDING:
001C9 COM_BUFF_COUNT:
001CA COM_I_SETCOUNT:
001CB COM_O_SETCOUNT:

001CC COM_I_STRUCNAME:
.BLKB 12
001D8 COM_O_STRUCNAME:
.BLKB 12
001E4 COM_O_BSRDATE:
.BLKB 8
001EC ALT_SSNAME:
.BLKB 32
0020C INPUT_FUNC:
.BLKB 1
0020D INPUT_RTYPE:
.BLKB 1
0020E OUTPUT_FUNC:
.BLKB 1
0020F FAST_STRUCLEV:
.BLKB 1
00210 INPUT_BEG:
.BLKB 0
00210 INPUT_CHAN:
.BLKB 4
00214 INPUT_FLAGS:
.BLKB 2
00216 INPUT_PADDING:
.BLKB 2
00218 INPUT_FAB:
.BLKB 4
0021C INPUT_NAM:
.BLKB 4
00220 INPUT_BCB:
.BLKB 4
00224 INPUT_QUAL:
.BLKB 4
00228 INPUT_BAD:
.BLKB 4
0022C INPUT_BLOCK:
.BLKB 4
00230 INPUT_MAXBLOCK:
.BLKB 4
00234 INPUT_MEDIA_ID:
.BLRB 4
00238 INPUT_NAMEDESC:
.BLKB 8
00240 INPUT_STATBLK:
.BLKB 8
00248 INPUT_HDR_BEG:
.BLKB 0
00248 INPUT_CREDATE:
.BLKB 8
00250 INPUT_REVDATE:
.BLKB 8
00258 INPUT_EXPDATE:
.BLKB 8
00260 INPUT_BAKDATE:
.BLKB 8
00268 INPUT_FILEOWNER:
.BLKB 4
0026C INPUT_FILECHAR:

00270 INPUT_RECATTR: .BLKB 4
00290 INPUT_HDR END: .BLKB 32
00290 INPUT_END: .BLKB 0
00290 INPUT_PROC_LIST: .BLKB 0
00294 INPUT_PLACEMENT: .BLKB 4
0029C INPUT_VBN_LIST: .BLKB 8
002A4 INPUT_PLACE_LEN: .BLKB 8
002A6 INPUT_PADDING_2: .BLKB 2
002A8 OUTPUT_BEG: .BLKB 2
002A8 OUTPUT_CHAN: .BLKB 0
002AC OUTPUT_FLAGS: .BLKB 4
002AE OUTPUT_PADDING: .BLKB 2
002B0 OUTPUT_FAB: .BLKB 2
002B4 OUTPUT_NAM: .BLKB 4
002B8 OUTPUT_BCB: .BLKB 4
002BC OUTPUT_QUAL: .BLKB 4
002C0 OUTPUT_BAD: .BLKB 4
002C4 OUTPUT_BLOCK: .BLKB 4
002C8 OUTPUT_MAXBLOCK: .BLKB 4
002CC OUTPUT_DEVGEOM: .BLKB 8
002D4 OUTPUT_ATTBUF: .BLKB 144
00364 OUTPUT_END: .BLKB 0
00364 LIST_TOTFILES: .BLKB 4
00368 LIST_TOTSIZE: .BLKB 4
0036C VERIFY_FAB: .BLKB 4
00370 VERIFY_USE_COUNT: .BLKB 4
00374 VERIFY_QUAL: .BLKB 4
00378 COMPARE_BCB: .BLKB 4

0037C FAST_BUFFER:
00380 FAST_BUFFER_SIZE:
00384 FAST_RVN:
00385 FAST_PADDING:
00386 DIR_VERLIMIT:
00388 FAST_VOL_BEG:
00388 FAST_IMAP_SIZE:
0038C FAST_IMAP:
00390 FAST_HDR_OFFSET:
00394 FAST_BOOT_LBN:
00398 FAST_VOL_END:
00398 JOUR_BUFFER:
0039C JOUR_DIR:
003A0 JOUR_HIBLK:
003A4 JOUR_EFBLK:
003A8 JOUR_INBLK:
003AC JOUR_FFBYTE:
003AE JOUR_INBYTE:
003B0 JOUR_STRUCTLEV:
003B2 JOUR_COUNT:
003B3 JOUR_REVERSE:
003B4 JOUR_EXSZ:
003B6 JOUR_PADDING:
003B8 CHKPTE_HIGH_SP:
003BC CHKPTE_LOW_SP:
003C0 CHKPTE_STACK:
003C4 CHKPTE_VARS:
003C8 CHKPTE_STATUS:
003CC DIR_BEG:.BLKB 0

003CC DIR_CHAN:
003D0 DIR_NAM:.BLKB 4
003D4 DIR_DEV_DESC:
003D8 DIR_SEL_DIR:
003E0 DIR_SEL_NTV:
003E8 DIR_STRUCLEV:
003E9 DIR_LEVELS:
003EA DIR_FLAGS:
003EB DIR_STATUS:
003EC DIR_STRING:
0052C DIR_STACK:
00790 DIR_SP:.BLKB 612
00794 DIR_SEL_LATEST:
00798 DIR_END:.BLKB 0
00798 DIR_SCANLIMIT:
007BC INPUT_MTL:
007C0 OUTPUT_MTL:
007C4 CURRENT_MTL:
007C8 CURRENT_VCB:
007CC CURRENT_WCB:
007D0 ACL_FIB_DESCR:
007D8 ACL_FIB:.BLKB 64
00818 ACL_LENGTH:
0081C ACL_BUFFER:
00820 CRYP_IN_CONTEXT:
00824 CRYP_OU_CONTEXT:
00828 CRYP_DA_CONTEXT:
0082C CRYP_DATA_ENCIV:
00834 CRYP_DATA_CODE:
00838 CRYP_DATA_KEY:
00840 CRYP_DATA_IV:

00000	P.AAB:	.ASCII	<5>\False\
00006	P.AAC:	.ASCII	<4>\True\
0000B		.BLKB	1
0000C	P.AAA:	.ADDRESS	P.AAB, P.AAC
00014	P.AAE:	.ASCII	<6>\SSNAME\
0001B	P.AAF:	.ASCII	<7>\COMMAND\
00023	P.AAG:	.ASCII	<7>\COMMENT\
0002B	P.AAH:	.ASCII	<8>\USERNAME\
00034	P.AAI:	.ASCII	<7>\USERUIC\
0003C	P.AAJ:	.ASCII	<4>\DATE\
00041	P.AAK:	.ASCII	<5>\OPSYS\
00047	P.AAL:	.ASCII	<6>\SYSVER\
0004E	P.AAM:	.ASCII	<8>\NODENAME\
00057	P.AAN:	.ASCII	<3>\SIR\
0005B	P.AAO:	.ASCII	<7>\DRIVEID\
00063	P.AAP:	.ASCII	<7>\BACKVER\
0006B	P.AAQ:	.ASCII	<9>\BLOCKSIZE\
00075	P.AAR:	.ASCII	<7>\XORSIZE\
0007D	P.AAS:	.ASCII	<7>\BUFFERS\
00085	P.AAT:	.ASCII	<9>\VOLSETNAM
0008F	P.AAU:	.ASCII	<5>\NVOLS\
00095	P.AAV:	.ASCII	<8>\BACKSIZE\
0009E	P.AAW:	.ASCII	<9>\BACKFILES
000A8	P.AAX:	.ASCII	<9>\VOLSTRUCT\
000B2	P.AAY:	.ASCII	<7>\VOLNAME\
000BA	P.AAZ:	.ASCII	<9>\OWNERNAME\
000C4	P.ABA:	.ASCII	<6>\FORMAT\
000CB	P.ABB:	.ASCII	<3>\RVN\
000CF	P.ABC:	.ASCII	<8>\VOOWNER\
000D8	P.ABD:	.ASCII	<7>\PROTECT\
000E0	P.ABE:	.ASCII	<8>\FILEPROT\
000E9	P.ABF:	.ASCII	<7>\RECPROT\
000F1	P.ABG:	.ASCII	<7>\VOLCHAR\
000F9	P.ABH:	.ASCII	<7>\VOLDATE\
00101	P.ABI:	.ASCII	<6>\WINDOW\
00108	P.ABJ:	.ASCII	<7>\LRU LIM\
00110	P.ABK:	.ASCII	<6>\EXTEND\
00117	P.ABL:	.ASCII	<7>\CLUSTER\
0011F	P.ABM:	.ASCII	<8>\RESFILES\
00128	P.ABN:	.ASCII	<7>\VOLSIZE\
00130	P.ABO:	.ASCII	<7>\TOTSIZE\
00138	P.ABP:	.ASCII	<8>\TOTFILES\
00141	P.ABQ:	.ASCII	<8>\MAXFILES\
0014A	P.ABR:	.ASCII	<9>\MAXFILENUM
00154	P.ABS:	.ASCII	<9>\SERIALNUM
0015E	P.ABT:	.ASCII	<8>\FILENAME\
00167	P.ABU:	.ASCII	<8>\STRUCTLEV\
00170	P.ABV:	.ASCII	<3>\FID\
00174	P.ABW:	.ASCII	<8>\BACKLINK\
0017D	P.ABX:	.ASCII	<8>\FILESIZE\
00186	P.ABY:	.ASCII	<3>\UIC\

13

4C	45	56	4F	52	50	46	04	0018A	P.ABZ:	.ASCII	<4>\FPRO\								
4E	52	54	52	41	48	43	55	05	0018F	P.ACA:	.ASCII	<4>\RPRO\							
	4F	49	53	41	43	45	52	07	00194	P.ACB:	.ASCII	<7>\ACLEVEL\							
	45	54	41	44	45	52	43	07	001A2	P.ACD:	.ASCII	<5>\UCHAR\							
	45	54	41	44	56	45	52	07	001AA	P.ACE:	.ASCII	<7>\RECATTR\							
	45	54	41	44	50	58	45	07	001B3	P.ACF:	.ASCII	<7>\CREDATE\							
	45	54	41	44	48	41	42	07	001BB	P.ACG:	.ASCII	<7>\REVDATE\							
	53	52	4F	54	43	45	53	07	001C3	P.ACH:	.ASCII	<7>\EXPDATE\							
	53	52	48	43	41	52	54	06	001CB	P.ACJ:	.ASCII	<7>\BAKDATE\							
	53	53	48	43	41	52	54	06	001D3	P.ACJ:	.ASCII	<7>\SECTORS\							
53	52	45	44	4E	49	4C	59	03	001E2	P.ACJ:	.ASCII	<6>\TRACKS\							
	4B	43	4F	4C	42	58	41	0D	001EC	P.ACJ:	.ASCII	<9>\CYLINDERS\							
			50	59	54	56	45	04	001F5	P.ACJ:	.ASCII	<8>\MAXBLOCK\							
			4C	41	49	52	45	04	001FC	P.ACJ:	.ASCII	<6>\DEVTYPE\							
			4D	41	4E	56	45	04	00203	P.ACJ:	.ASCII	<6>\SERIAL\							
			4C	45	42	41	4C	05	0020A	P.ACJ:	.ASCII	<6>\DEVNAME\							
			4B	43	4F	4C	42	04	00210	P.ACJ:	.ASCII	<5>\LABEL\							
			4F	42	4C	58	45	04	00219	P.ACJ:	.ASCII	<8>\BADBLOCK\							
4B	43	4F	4C	42	54	4F	4E	08	00222	P.ACJ:	.ASCII	<8>\INDEXLBN\							
			4E	42	56	54	4F	07	0022C	P.ACJ:	.ASCII	<9>\BOOTBLOCK\							
54	4E	45	4D	45	43	41	4C	09	00234	P.ACJ:	.ASCII	<9>\PLACEMENT\							
			43	49	55	5F	52	09	0023E	P.ACJ:	.ASCII	<7>\DIR_UIC\							
			4F	52	50	46	5F	09	00246	P.ACJ:	.ASCII	<8>\DIR_FPRO\							
53	55	54	41	54	53	5F	52	09	0024F	P.ACJ:	.ASCII	<10>\DIR_STATUS\							
4D	49	4C	52	45	56	5F	52	09	0025A	P.ACJ:	.ASCII	<10>\DIR_VERLIM\							
			54	49	4D	49	4C	08	00265	P.ACJ:	.ASCII	<8>\VERLIMIT\							
			4E	49	4D	4E	49	09	0026E	P.ACJ:	.ASCII	<9>\RETAINMIN\							
			58	41	4D	4E	49	09	00278	P.ACJ:	.ASCII	<9>\RETAINMAX\							
54	4E	45	4D	47	45	53	4C	0A	00282	P.ACJ:	.ASCII	<10>\ACLSEGMENT\							
			52	45	54	41	57	09	0028D	P.ACJ:	.ASCII	<9>\HIGHWATER\							
			53	47	41	4C	46	09	00297	P.ACJ:	.ASCII	<9>\JNL_FLAGS\							
			59	45	4B	54	41	0A	002A1	P.ACJ:	.ASCII	<10>\CRPDATAKEY\							
00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	002AC	P.ACJ:	.ADDRESS	P.AAE, P.AAF, P.AAG, P.AAH, P.AAI, -								
00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	002C4			P.AAJ, P.AAK, P.AAL, P.AAM, P.AAN, P.AAO, -								
00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	002DC			P.AAP, P.AAQ, P.AAR, P.AAS, P.AAT, P.AAU, -								
00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	002F4			P.AAV, P.AAW, P.AAX, P.AAY, P.AAZ, P.ABA, -								
00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	0030C			P.ABB, P.ABC, P.ABD, P.ABE, P.ABF, P.ABG, -								
00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00324			P.ABH, P.ABI, P.ABJ, P.ABK, P.ABL, P.ABM, -								
00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	0033C			P.ABN, P.ABO, P.ABP, P.ABQ, P.ABR, P.ABS, -								
00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00354			P.ABT, P.ABU, P.ABV, P.ABW, P.ABX, P.ABY, -								
00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	0036C			P.ABZ, P.ACA, P.AC, P.AC, P.AC, P.AC, -								
00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00384			P.ACF, P.ACG, P.AC, P.AC, P.AC, P.AC, -								
00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	0039C			P.AC, P.AC, P.AC, P.AC, P.AC, P.AC, -								
00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	003B4			P.AC, P.AC, P.AC, P.AC, P.AC, P.AC, -								
00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	003CC			P.AC, P.AC, P.AC, P.AC, P.AC, P.AC, -								
								003E4			P.AC, P.AC, P.AC, P.AC, P.AC, P.AC, -								
53	33	21	20	3D	20	45	5A	49	53	20	20	20	20	1B	003F0	P.ACJ:	.ASCII	<27>\ SIZE = !3SL, TYPE = !AC\	
								20	20	20	20	20	20	20	06	0040C	P.ACJ:	.ASCII	<6>\
								22	46	41	21	22	05	03	00413	P.ACJ:	.ASCII	<5>\!"AF"\	
									55	25	21	03	03	00419	P.ACJ:	.ASCII	<3>\!XU\		
									44	25	21	03	03	0041D	P.ACJ:	.ASCII	<3>\!ZD\		
									4C	55	21	03	03	00421	P.ACJ:	.ASCII	<3>\!UL\		
									4C	55	21	03	03	00425	P.ACJ:	.ASCII	<3>\!UL\		
29	46	41	21	2C	46	41	21	28	20	4C	58	21	0D	00429	P.ACJ:	.ASCII	<13>\!XL (!AF,!AF)\		

4C 55 21 2C 4C 55 21 2C 4C 55 21 0B 00437 P.ADP: .ASCII <11>!\UL, !UL, !UL\	4C 55 21 3A 4C 55 21 08 00443 P.ADG: .ASCII <8>!\UL: !UL \	4C 4F 21 03 0044C P.ADR: .ASCII <3>!\OL\
0D 00450 P.ATD: .ASCII <13>	0A 00451 .ASCII <10>	00452 .BLKB 2
00000002, 00454 P.ADS: .LONG 2	00000000, 00459 .ADDRESS P.ATD	0045C P.ADU: .ASCII <3>!\AS\
68 74 69 77 20 64 65 74 70 79 72 53 41 21 03 00460 P.ADV: .ASCII <29>\Encrypted with algorithm: !XB\	42 58 21 20 3A 60 68 74 69 72 6F 67 6C 61 20 0046F .ADDRES	49 20 2C 4C 58 21 20 4C 58 21 20 4C 58 21 3A 76 0047E P.ADW: .ASCII <23>\Key: !XL !XL, Iv: !XL !XL\
42 58 21 20 3A 60 68 74 69 72 6F 67 6C 61 20 0048D .ADDX: .ASCII <3>!\XB\		

FALSETRUE=	P.AAA
ATTRS=	P.AAD
.EXTRN	DEBLOCK, DEBLOCK_ATTR
.EXTRN	DECODE_DEVTYPE, FIN_IN_SAVE
.EXTRN	INIT_IN_SAVE, LIST_FA0
.EXTRN	LIST_EOL, LIST_PROTECTION
.EXTRN	READ_BUFFER, RESTORE_HANDLER
.EXTRN	CRYPTO_CHKSAV, BACKUPS_BACNOTENC
.EXTRN	BACKUPS_ENCSAVSET
.EXTRN	SYSSFORMAT_ACL
.WEAK	CRYPTO_INIDER, CRYPTO_DECR_BLOCK

07FC 00000 ANALYZE_ONE_ATTRIBUTE:

5A 0000000G 00 9E 0002	.WORD Save R2, R3, R4, R5, R6, R7, R8, R9, R10	1727
5E FDE0 CE 9E 0009	MOVAB LIST_FA0, R10	
56 04 AC D0 000E	MOVAB -544TSP), SP	1852
50 02 A6 3C 00012	MOVL ATT, R6	
57 FDF3 CF40 DD 00016	MOVZWL 2(R6), R0	
57 66 3C 0001B	PUSHL ATTRS-4[R0]	
57 57 DD 0001E	MOVZWL (R6), R7	
FF32 CF 9F 00020	PUSHL R7	
6A 03 FB 00024	PUSHAB P.ADH	
0000000G 00 FF40 CF 9F 00027	CALLS #3, LIST_FA0	1853
6A 00 FB 00027	CALLS #0, LIST_EOL	1858
6A 01 FB 00032	PUSHAB P.ADI	1859
0051 8F 01 02 A6 AF 00035	CALLS #1, LIST_FA0	
008A 008A 008A 008A 008A 0003C 1\$:	CASEW 2(R6), #T, #81	
008A 00A4 00D1 00C8 00044 .WORD	5\$-1\$,-	
008A 00BA 00A4 00BA 0004C	5\$-1\$,-	
008A 00DA 00DA 00DA 00054	5\$-1\$,-	
00A4 00DA 00A4 00DA 0005C	7\$-1\$,-	
00DA 008A 00BA 00BA 00064	8\$-1\$,-	
016D 0165 015D 00C8 0006C	2\$-1\$,-	
00DA 00DA 00D1 00A4 00074	5\$-1\$,-	
00DA 00DA 00DA 00DA 0007C	5\$-1\$,-	
00DA 00DA 00DA 00A4 00084	2\$-1\$,-	
013D 00A4 00BA 019D 0008C	5\$-1\$,-	
0165 00C8 00DA 013D 00094	5\$-1\$,-	
00A4 00A4 00A4 016D 0009C	9\$-1\$,-	
00D1 00D1 00D1 00DA 000A4	9\$-1\$,-	
00DA 00DA 00DA 00D1 000AC	9\$-1\$,-	

ANALYZE
V04-000

Analyze a save set
ANALYZE_ONE_ATTRIBUTE - analyze contents of att

K 3

15-Sep-1984 23:40:04
14-Sep-1984 11:53:45

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]ANALYZE.B32;1

Page 19
(3)

008A	019D	00EA	00DA	000B4	5\$-1\$,-
00A4	00DA	0180	00BA	000BC	9\$-1\$,-
0165	00C8	00A4	00DA	000C4	2\$-1\$,-
00D1	00DA	00DA	00A4	000CC	9\$-1\$,-
00A4	00DA	01AA	00D1	000D4	2\$-1\$,-
		00A4	0227	000DC	5\$-1\$,-
					5\$-1\$,-
					5\$-1\$,-
					9\$-1\$,-
					7\$-1\$,-
					15\$-1\$,-
					16\$-1\$,-
					17\$-1\$,-
					2\$-1\$,-
					8\$-1\$,-
					9\$-1\$,-
					9\$-1\$,-
					9\$-1\$,-
					9\$-1\$,-
					9\$-1\$,-
					9\$-1\$,-
					2\$-1\$,-
					9\$-1\$,-
					9\$-1\$,-
					9\$-1\$,-
					21\$-1\$,-
					5\$-1\$,-
					2\$-1\$,-
					14\$-1\$,-
					14\$-1\$,-
					9\$-1\$,-
					7\$-1\$,-
					16\$-1\$,-
					17\$-1\$,-
					2\$-1\$,-
					2\$-1\$,-
					2\$-1\$,-
					9\$-1\$,-
					8\$-1\$,-
					8\$-1\$,-
					8\$-1\$,-
					8\$-1\$,-
					9\$-1\$,-
					9\$-1\$,-
					9\$-1\$,-
					9\$-1\$,-
					10\$-1\$,-
					21\$-1\$,-
					5\$-1\$,-
					5\$-1\$,-
					19\$-1\$,-
					9\$-1\$,-
					2\$-1\$,-
					9\$-1\$,-
					2\$-1\$,-
					7\$-1\$,-
					16\$-1\$,-

		50	04	A6	3C 00181	MOVZWL	4(R6), R0	
		52	09	A6	9A 00185	MOVZBL	9(R6), R2	
			10	78	00189	ASHL	#16, R2, R2	
			6240	9F	0018D	PUSHAB	(R2)[R0]	
		6A	04	CF	9F 00190	PUSHAB	P.ADP	
			04	FB	00194	CALLS	#4 LIST_FA0	
			4A	11	00197	BRB	23\$	
			44435752	8F	DD 00199	PUSHL	#1145263954	1928
			0E	11	0019F	BRB	18\$	
			44455752	8F	DD 001A1	PUSHL	#1145395026	1931
			06	11	001A7	BRB	18\$	
		00000000G	44555752	8F	DD 001A9	PUSHL	#1146443602	1934
			04	A6	3C 001AF	MOVZWL	4(R6), -(SP)	
			02	FB	001B3	CALLS	#2 LIST_PROTECTION	
			27	11	001BA	BRB	23\$	
			52	04	A6 9E 001BC	MOVAB	4(R6), P	1938
			50	04	A746 9E 001C0	MOVAB	4(R7)[R6], R0	1939
			50	52	D1 001C5	CMPL	P, R0	
			39	1E	001C8	BGEQU	26\$	
			7E	62	7D 001CA	MOVQ	(P), -(SP)	1941
			FDD8	CF	9F 001CD	PUSHAB	P.ADP	
			6A	03	FB 001D1	CALLS	#3, LIST_FA0	
			52	08	CO 001D4	ADDL2	#8, P	1942
				E7	11 001D7	BRB	20\$	1939
			FDD2	04	A6 DD 001D9	PUSHL	4(R6)	1948
			6A	02	FB 001E0	PUSHAB	P.ADR	
			00A8	31	001E3	CALLS	#2 LIST_FA0	
			59	04	A6 9E 001E6	BRW	29\$	
			58	59	D0 001EA	MOVAB	4(R6), R9	1957
08	00	6E	6E	00	2C 001ED	MOVL	R9, ACE_POINTER	
08	00	6E	F8	AD	001F2	MOVCS	#0, (SP), #0, #8, ACE_BINDESC	1958
			6E	00	2C 001F4	MOVCS	#0, (SP), #0, #8, ACE_TXTDESC	1959
			F0	AD	001F9			
			50	04	A746 9E 001FB	MOVAB	4(R7)[R6], R0	1960
			50	58	D1 00200	CMPL	ACE_POINTER, R0	
				DE	1E 00203	BGEQU	23\$	
		F8	AD	68	9B 00205	MOVZBW	(ACE_POINTER), ACE_BINDESC	1963
		FC	AD	58	D0 00209	MOVL	ACE_POINTER, ACE_BINDESC+4	1964
		FO	AD	0200	8F B0 0020D	MOVW	#512, ACE_TXTDESC	1965
		F4	AD	10	AE 9E 00213	MOVAB	ACE_TEXT, ACE_TXTDESC+4	1966
				7E	D4 00218	CLRL	-(SP)	1972
		08	AE	06	D0 0021A	MOVL	#6, 8(SP)	
				08	AE 9F 0021E	PUSHAB	8(SP)	
		0C	AE	FD95	CF 9F 00221	PUSHAB	P.ADS	
			50	8F	9A 00225	MOVZBL	#80, 12(SP)	
			0C	AE	9F 0022A	PUSHAB	12(SP)	
			FO	AD	9F 0022D	PUSHAB	ACE_TXTDESC	
			FO	AD	9F 00230	PUSHAB	ACE_TXTDESC	
		00000000G	00	F8	AD 9F 00233	PUSHAB	ACE_BINDESC	
			59	07	FB 00236	CALLS	#7, SYS\$FORMAT_ACL	
				58	D1 00230	CMPL	ACÉ_POINTER, R9	1978
				C8	12 00240	BNEQ	27\$	
		F0	AD	06	A2 00242	SUBW2	#6, ACE_TXTDESC	1981
		F4	AD	06	CO 00246	ADDL2	#6, ACE_TXTDESC+4	1982
			FD71	AD	9F 0024A	PUSHAB	ACÉ_TXTDESC	1984
				CF	9F 0024D	PUSHAB	P.ADU	

00000000G	6A 00	02 FB 00251	CALLS #2, LIST_FA0	1985
	50	00 FB 00254	CALLS #0, LIST_EOL	1986
	58	68 9A 0025B	MOVZBL (ACE_POINTER) R0	
	7E	50 C0 0025E	ADDL2 R0 ACE_POINTER	
		98 11 00261	BRB 25\$	
		66 9A 00263	28\$: MOVZBL (R6), -(SP)	1960
		CF 9F 00266	PUSHAB P,ADV	1992
00000000G	6A 00	02 FB 0026A	CALLS #2, LIST_FA0	1993
		00 FB 0026D	CALLS #0, LIST_EOL	1994
		66 95 00274	TSTB (R6)	
	7E	16 13 00276	BEQL 29\$	
	7E	0C A6 7D 00278	MOVQ 12(R6), -(SP)	2001
	04	A6 7D 0027C	MOVQ 4(R6), -(SP)	
		FD60 CF 9F 00280	PUSHAB P,ADV	
00000000G	6A 00	05 FB 00284	CALLS #5, LIST_FA0	2002
00000000G	00	00 FB 00287	CALLS #0, LIST_EOL	2013
		00 FB 0028E	29\$: CALLS #0, LIST_EOL	
		04 00295	RET	2014

: Routine Size: 662 bytes, Routine Base: CODE + 049A

```
: 464 2015 1 %SBTTL 'ANALYZE_ONE_RECORD - analyze save set record'
: 465 2016 1 ROUTINE ANALYZE_ONE_RECORD(REC): L_PS NOVALUE=
: 466 2017 1
: 467 2018 1 ++
: 468 2019 1
: 469 2020 1 FUNCTIONAL DESCRIPTION:
: 470 2021 1 This routine analyzes the contents of one save set record.
: 471 2022 1
: 472 2023 1 INPUT PARAMETERS:
: 473 2024 1 REC - Pointer to record.
: 474 2025 1
: 475 2026 1 IMPLICIT INPUTS:
: 476 2027 1 NONE
: 477 2028 1
: 478 2029 1 OUTPUT PARAMETERS:
: 479 2030 1 NONE
: 480 2031 1
: 481 2032 1 IMPLICIT OUTPUTS:
: 482 2033 1 NONE
: 483 2034 1
: 484 2035 1 ROUTINE VALUE:
: 485 2036 1 NONE
: 486 2037 1
: 487 2038 1 SIDE EFFECTS:
: 488 2039 1 The listing is produced.
: 489 2040 1
: 490 2041 1 --
: 491 2042 1
: 492 2043 2 BEGIN
: 493 2044 2 MAP
: 494 2045 2 REC: REF BBLOCK; ! Pointer to record
: 495 2046 2 BIND
: 496 2047 2 RTYPES = UPLIT (
: 497 2048 2 UPLIT BYTE (%ASCIC 'NULL'),
: 498 2049 2 UPLIT BYTE (%ASCIC 'SUMMARY'),
: 499 2050 2 UPLIT BYTE (%ASCIC 'VOLUME'),
: 500 2051 2 UPLIT BYTE (%ASCIC 'FILE'),
: 501 2052 2 UPLIT BYTE (%ASCIC 'VBN'),
: 502 2053 2 UPLIT BYTE (%ASCIC 'PHYSVOL'),
: 503 2054 2 UPLIT BYTE (%ASCIC 'LBN'),
: 504 2055 2 UPLIT BYTE (%ASCIC 'FID'),
: 505 2056 2 UPLIT BYTE (%ASCIC 'FILE_EXT'))
: 506 2057 2 : VECTOR:
: 507 2058 2 L_DECL:
: 508 2059 2
: 509 2060 2
: 510 2061 2 : Format the record header.
: 511 2062 2
: 512 2063 2 FAO_('Record header');
: 513 2064 2 EOL();
: 514 2065 2 FAO_(' RSIZE = !UL!- = %X'!XW'', .REC[BRHSW_RSIZE]);
: 515 2066 2 EOL();
: 516 2067 2 FAO_(' RTYPE = !AC', .RTYPES[.REC[BRHSW_RTYPE]]);
: 517 2068 2 EOL();
: 518 2069 2 FAO_(' BADDATA = !AC', .FALSETRUE[.REC[BRHSV_BADDATA]]);
: 519 2070 2 EOL();
: 520 2071 2 FAO_(' DIRECTORY = !AC', .FALSETRUE[.REC[BRHSV_DIRECTORY]]);
```

```

: 521 2072 2 EOL_(); 2073 2 FAO_(' ADDRESS = !UL', .REC[BRH$L_ADDRESS]); 2074 2 EOL_(); 2075 2 EOL_(); 2076 2
: 522 2077 2
: 523 2078 2 ! Format the record contents. 2079 2
: 524 2080 2 CASE .REC[BRH$W_RTYPE] FROM BRH$K_NULL TO BRH$K_FILE_EXT OF 2081 2 SET 2082 2
: 525 2083 2 [BRH$K_NULL, BRH$K_VBN, BRH$K_LBN, OUTRANGE]: 2084 2 0;
: 526 2085 2 [BRH$K_SUMMARY, BRH$K_VOLUME, BRH$K_FILE, BRH$K_FILE_EXT, BRH$K_PHYSVOL]: 2086 2
: 527 2087 2 BEGIN 2088 2 FAO_(' STRUCLEV = !XW', 2089 2 .BBLOCK[REC[BRH$C_LENGTH,0,0,0], BSASW_STRUCLEV]); 2090 2 EOL_(); 2091 2 DEB[OCK_ATTR(.REC, 0, ANALYZE_ONE_ATTRIBUTE); 2092 2 EOL_(); 2093 2 END;
: 528 2094 2 [BRH$K_FID]: 2095 2 BEGIN 2096 2
: 529 2097 2 FAO_(' STRUCLEV = !XW', 2098 2 .BBLOCK[REC[BRH$C_LENGTH,0,0,0], BSASW_STRUCLEV]); 2099 2 EOL_(); 2100 2 FAO_(' FID COUNT = !UL', 2101 2 .BBLOCK[REC[BRH$C_LENGTH,0,0,0], BSASW_FID_COUNT]); 2102 2 EOL_(); 2103 2 INCR I FROM 0 TO .BBLOCK[REC[BRH$C_LENGTH,0,0,0], BSASW_FID_COUNT]-1 DO 2104 2 BEGIN 2105 2 FAO_(' FID = (!UL,!UL,!UL)', 2106 2 .BBLOCK[REC[BRH$C_LENGTH,0,0,0], BSASW_FID_NUM] + 2107 2 .BBLOCK[REC[BRH$C_LENGTH,0,0,0], BSASB_FID_NMX] ^ 16 + .I, 2108 2 .BBLOCK[REC[BRH$C_LENGTH+ i*2,0,0,0], BSA$W_FID_SEQ], 2109 2 .BBLOCK[REC[BRH$C_LENGTH,0,0,0], BSASB_FID_RVN]); 2110 2 EOL_(); 2111 2 END;
: 530 2112 2 EOL_(); 2113 2 END;
: 531 2114 2
: 532 2115 2 TES: 2116 1 END;

```

59	52	4:	4C	4C	55	4E	04	00730	P.ADZ:	.ASCII	<4>\NULL\	
45	4D		55	4C	55	53	07	00735	P.AEA:	.ASCII	<7>\SUMMARY\	
			45	4C	4F	56	06	0073D	P.AEB:	.ASCII	<6>\VOLUME\	
				45	4C	49	46	04	00744	P.AEC:	.ASCII	<4>\FILE\
					4E	42	56	03	00749	P.AED:	.ASCII	<3>\VBN\
					4E	42	4C	03	0074D	P.AEE:	.ASCII	<7>\PHYSVOL\
					4E	49	46	03	00755	P.AEF:	.ASCII	<3>\LBN\
					44	49	46	03	00759	P.AEG:	.ASCII	<3>\FID\
54	58	45	5F	45	4C	49	46	08	0075D	P.AEH:	.ASCII	<8>\FILE_EXT\

RTYPES = P-ARY

03FC 00000 ANALYZE ONE RECORD:

35C 00000 ANALYZE ONE RECORD:									
								WORD	Save R2,R3,R4,R5,R6,R7,R8,R9
								MOVAB	FALSETRUE, R9
								MOVAB	LIST_FA0, R8
								MOVAB	LIST_EOL, R7
								PUSHAB	P.AEJ
								CALLS	#1, LIST_FA0
								CALLS	#0, LIST_EOL
								MOVL	REC, R4
								MOVZWL	(R4), -(SP)
								PUSHAB	P.AEJ
								CALLS	#2, LIST_FA0
								CALLS	#0, LIST_EOL
								MOVZWL	2(R4), R2
								PUSHL	RTYPE[R2]
								PUSHAB	P.AEK
								CALLS	#2, LIST_FA0
								CALLS	#0, LIST_EOL
								EXTZV	#0, #1, Z(R4), R0
								PUSHL	FALSETRUE[R0]
								PUSHAB	P.AEL
								CALLS	#2, LIST_FA0
								CALLS	#0, LIST_EOL
								EXTZV	#1, #1, Z(R4), R0
								PUSHL	FALSETRUE[R0]
								PUSHAB	P.AEM
								CALLS	#2, LIST_FA0
								CALLS	#0, LIST_EOL
								PUSHL	8(R4)
								PUSHAB	P.AEN
								CALLS	#2, LIST_FA0

		67	00	FB	00073	CALLS	#0, LIST_EOL	2074
		67	00	FB	00076	CALLS	#0, LIST_EOL	2075
0013	08	0013	52	AF	00079	CASEW	R2, #0, #8	2080
0032	0013	0013	0089	0007D	1\$:	WORD	7\$-1\$,-	
		0089	0089	00085			2\$-1\$,-	
		0013	0013	0008D			2\$-1\$,-	
							7\$-1\$,-	
							2\$-1\$,-	
							7\$-1\$,-	
							2\$-1\$,-	
							7\$-1\$,-	
							3\$-1\$,-	
							2\$-1\$,-	
						RET		
		7E	10	A4	04 0008F	MOVZWL	16(R4), -(SP)	2089
			07F4	C9	3C 00090	PUSHAB	P.AEO	
		68	02	FB	00094	CALLS	#2, LIST_FA0	2090
		67	00	FB	00098	CALLS	#0, LIST_EOL	2091
			FBA6	CF	9F 0009E	PUSHAB	ANALYZE_ONE_ATTRIBUTE	
				7E	D4 000A2	CLRL	-(SP)	
		00000000G	00	54	DD 000A4	PUSHL	R4	
				54	11 000A6	CALLS	#3, DEBLOCK_ATTR	
				54	11 000AD	BRB	6\$	2092
		52	10	A4	9E 000AF	MOVAB	16(R4), R2	2098
		7E	62	3C	000B3	MOVZWL	(R2), -(SP)	
			0807	C9	9F 000B6	PUSHAB	P.AEP	
		68	02	FB	000BA	CALLS	#2, LIST_FA0	
		67	00	FB	000BD	CALLS	#0, LIST_EOL	
		7E	06	A2	3C 000C0	MOVZWL	6(R2), -(SP)	2101
			081A	C9	9F 000C4	PUSHAB	P.AEQ	
		68	02	FB	000C8	CALLS	#2, LIST_FA0	
		67	00	FB	000CB	CALLS	#0, LIST_EOL	
		56	06	A2	3C 000CE	MOVZWL	6(R2), R6	2102
		55	02	A2	9E 000D2	MOVAB	2(R2), R5	2103
		53	01	CE	C00D6	MNEG	#1, I	2109
				24	11 000D9	BRB	5\$	
		7E	04	A2	9A 000DB	MOVZBL	4(R2), -(SP)	
			18	A443	3C 000DF	MOVZWL	24(R4)[I], -(SP)	
		50	65	3C	000E4	MOVZWL	(R5), R0	
		51	05	A2	9A 000E7	MOVZBL	5(R2), R1	
		51	10	78	000EB	ASHL	#16, R1, R1	
		50	51	C0	000EF	ADDL2	R1, R0	
			6340	9F	000F2	PUSHAB	(I)[R0]	
			082E	C9	9F 000F5	PUSHAB	P.AER	
		68	04	FB	000F9	CALLS	#4, LIST_FA0	
		67	00	FB	000FC	CALLS	#0, LIST_EOL	
D8	53		56	F2	000FF	AOBLSS	R6, I, 4\$	2110
	67		00	FB	00103	CALLS	#0, LIS_EOL	2103
			04	00106	6\$:	RET		2112
					7\$:			2116

: Routine Size: 263 bytes, Routine Base: CODE + 0852

```
: 567 2117 1 %SBTTL 'ANALYZE_ONE_BUFFER - analyze save set buffer'
568 2118 1 ROUTINE ANALYZE_ONE_BUFFER(BCB): [_PS NOVALUE=
569
570 2120 1 !++
571 2121 1
572 2122 1 FUNCTIONAL DESCRIPTION:
573 2123 1 This routine analyzes the contents of one save set buffer.
574
575 2125 1 INPUT PARAMETERS:
576 2126 1 BCB - Pointer to buffer control block.
577
578 2128 1 IMPLICIT INPUTS:
579 2129 1 NONE
580
581 2131 1 OUTPUT PARAMETERS:
582 2132 1 NONE
583
584 2134 1 IMPLICIT OUTPUTS:
585 2135 1 NONE
586
587 2137 1 ROUTINE VALUE:
588 2138 1 NONE
589
590 2140 1 SIDE EFFECTS:
591 2141 1 The listing is produced.
592 2142 1 The buffer is released.
593
594 2144 1 !--
595
596 2146 2 BEGIN
597 2147 2 MAP
598 2148 2 LOCAL BCB: REF BBLOCK; ! Pointer to buffer control block
599 2149 2 LOCAL BUF: REF BBLOCK; ! Pointer to buffer
600
601 2151 2 L_DECL:
602
603 2153 2 ! Point to buffer.
604
605 2155 2 BUF = .BCB[BCB_BUFFER];
606
607 2157 2
608
609 2159 2 ! Format the block header.
610
611 2160 2 FAO_('Block header');
612 2161 2 EOL();
613 2162 2 FAO_(' SIZE = !UL', .BUF[BBH$W_SIZE]);
614 2163 2 EOL();
615 2164 2 FAO_(' OPSYS = !UL', .BUF[BBH$W_OPSYS]);
616 2165 2 EOL();
617 2166 2 FAO_(' SUBSYS = !UL', .BUF[BBH$W_SUBSYS]);
618 2167 2 EOL();
619 2168 2 FAO_(' APPLIC = !UL', .BUF[BBH$W_APPLIC]);
620 2169 2 EOL();
621 2170 2 FAO_(' NUMBER = !UL', .BUF[BBH$L_NUMBER]);
622 2171 2 EOL();
623 2172 2 FAO_(' STRUCLEV = 'XW', .BUF[BBH$W_STRUCLEV]);
```

```

624 2174 2 EOL();
625 2175 2 FAO(' VOLNUM = !UL', .BUF[BBH$W_VOLNUM]);
626 2176 2 EOL();
627 2177 2 FAO(' CRC = !XL', .BUF[BBH$L_CRC]);
628 2178 2 EOL();
629 2179 2 FAO(' BLOCKSIZE = !UL', .BUF[BBH$L_BLOCKSIZE]);
630 2180 2 EOL();
631 2181 2 FAO(' NOCRC = !AC', .FALSETRUE[.BUF[BBH$V_NOCRC]]);
632 2182 2 EOL();
633 2183 2 FAO(' SSNAME = "'!AC'", .BUF[BBH$T_SSNAME]);
634 2184 2 EOL();
635 P 2185 2 FAO(' FID = !UL,!UL,!UL
636 P 2186 2 .BUF[BBH$W_FID_NUM] + .BUF[BBH$B_FID_NMX] ^ 16,
637 P 2187 2 .BUF[BBH$W_FID_SEQ];
638 P 2188 2 .BUF[BBH$B_FID_RVN];
639 2189 2 EOL();
640 P 2190 2 FAO(' DID = !UL,!UL,!UL
641 P 2191 2 .BUF[BBH$W_DID_NUM] + .BUF[BBH$B_DID_NMX] ^ 16,
642 P 2192 2 .BUF[BBH$W_DID_SEQ];
643 P 2193 2 .BUF[BBH$B_DID_RVN];
644 2194 2 EOL();
645 2195 2 FAO(' FILENAME = "'!AC'", .BUF[BBH$T_FILENAME]);
646 2196 2 EOL();
647 2197 2 FAO(' ATTRIB = !XL!XL', .(BUF[BBH$B_BKTSIZE]), .(BUF[BBH$B_RTYPE]));
648 2198 2 EOL();
649 2199 2 FAO(' FILESIZE = !UL', .BUF[BBH$L_FILESIZE]);
650 2200 2 EOL();
651 2201 2 EOL();
652 2202 2 EOL();
653 2203 2 EOL();
654 2204 2 ! Format the records contained in this buffer.
655 2205 2 EOL();
656 2206 2 DEBLOCK(.BCB, ANALYZE_ONE_RECORD);
657 2207 1 END;

```

20	3D	72	65	64	61	65	68	20	68	63	6F	6C	42	0C	00959	P.AES:	.ASCII	<12>\Block header\		
		20	20	20	20	20	20	45	5A	49	53	20	20	11	00966	P.AET:	.ASCII	<17>\ SIZE	= !UL\	
		20	20	20	20	20	53	59	53	50	4F	20	20	11	00978	P.AEU:	.ASCII	<17>\ OPSYS	= !UL\	
		20	20	20	20	53	59	53	42	55	53	20	20	11	0098A	P.AEV:	.ASCII	<17>\ SUBSYS	= !UL\	
		20	20	20	20	43	49	4C	50	50	41	20	20	11	0099C	P.AEW:	.ASCII	<17>\ APPLIC	= !UL\	
		20	20	20	20	52	45	42	4D	55	4E	20	20	11	009AE	P.AEX:	.ASCII	<17>\ NUMBER	= !UL\	
		20	20	20	56	45	4C	43	55	52	54	53	20	20	11	009C0	P.AEY:	.ASCII	<17>\ STRUCLEV	= !XL\
		20	20	20	20	4D	55	4E	4C	4F	56	20	20	11	009D2	P.AEZ:	.ASCII	<17>\ VOLNUM	= !UL\	
		20	20	20	20	20	20	43	52	43	20	20	11	009E4	P.AFA:	.ASCII	<17>\ CRC	= !XL\		
		20	3D	45	5A	49	53	4B	43	4F	4C	42	20	20	11	009F6	P.AFB:	.ASCII	<17>\ BLOCKSIZE	= !UL\
		20	20	20	20	43	52	43	4F	4E	20	20	11	00A05	P.AFC:	.ASCII	<17>\ NOCRC	= !AC\		

007C 00000 ANALYZE_ONE_BUFFER:

007C 00000 ANALYZE ONE BUFFER.						.WORD	Save R2,R3,R4,R5,R6
56	FEB6	CF	9E	00002		MOVAB	P.AES, R6
55	00000000G	00	9E	00007		MOVAB	LIST_FA0, R5
54	00000000G	00	9E	0000E		MOVAB	LIST_EOL, R4
53	04	A5	00	00015		MOVL	BCB, R3
52	0C	A3	00	00019		MOVL	12(R3), BUF
		56	DD	0001D		PUSHL	R6
65		01	FB	0001F		CALLS	#1, LIST_FA0
64		00	FB	00022		CALLS	#0, LIST_EOL
7E		62	3C	00025		MOVZWL	(BUF), -(SP)
	0D	A6	9F	00028		PUSHAB	P.AET
65		02	FB	0002B		CALLS	#2, LIST_FA0
64		00	FB	0002E		CALLS	#0, LIST_EOL
7E	02	A2	3C	00031		MOVZWL	2(BUF), -(SP)
	1F	A6	9F	00035		PUSHAB	P.AEU
65		02	FB	00038		CALLS	#2, LIST_FA0
64		00	FB	0003B		CALLS	#0, LIST_EOL
7E	04	A2	3C	0003E		MOVZWL	4(BUF), -(SP)
	31	A6	9F	00042		PUSHAB	P.AEV
65		02	FB	00045		CALLS	#2, LIST_FA0
64		00	FB	00048		CALLS	#0, LIST_EOL
7E	06	A2	3C	0004B		MOVZWL	6(BUF), -(SP)
	43	A6	9F	0004F		PUSHAB	P.AEW
65		02	FB	00052		CALLS	#2, LIST_FA0
64		00	FB	00055		CALLS	#0, LIST_EOL
	08	A2	DD	0005C		PUSHL	8(BUF)
	55	A6	9F	0005B		PUSHAB	P.AEX
65		02	FB	0005E		CALLS	#2, LIST_FA0
64		00	FB	00061		CALLS	#0, LIST_EOL
7E	20	A2	3C	00064		MOVZWL	32(BUF), -(SP)
	67	A6	9F	00058		PUSHAB	P.AEY
65		02	FB	0006B		CALLS	#2, LIST_FA0
64		00	FB	0006E		CALLS	#0, LIST_EOL
7E	22	A2	3C	00071		MOVZWL	34(BUF), -(SP)
	79	A6	9F	00075		PUSHAB	P.AEZ
65		02	FB	00078		CALLS	#2, LIST_FA0
64		00	FB	0007B		CALLS	#0, LIST_EOL
	24	A2	DD	0007E		PUSHL	36(BUF)
008B	C6	9F	00081		PUSHAB	P.AFA	
65	02	FB	00085		CALLS	#2, LIST_FA0	

50	20	A2	64	00	FB 00088	CALLS #0, LIST_EOL	2178
			64	28 009D	A2 DD 0008B	PUSHL 40(BUF)	2179
			65	02	C6 9F 0008E	PUSHAB P.AFB	
			64	00	FB 00092	CALLS #2, LIST_FA0	2180
			65	00	EF 00095	CALLS #0, LIST_EOL	2181
			64	00	DD 0009E	EXTZV #0, #1, 4(BUF), R0	
			65	02	C6 9F 000A3	PUSHL FALSETRUE[R0]	
			64	00	FB 000A7	PUSHAB P.AFC	
			65	30 00C1	A2 9F 000AD	CALLS #2, LIST_FA0	2182
			64	00	C6 9F 000B0	CALLS #0, LIST_EOL	2183
			65	02	FB 000B4	PUSHAB 48(BUF)	
			64	00	FB 000B7	PUSHAB P.AFD	
			7E	54	A2 9A 000BA	CALLS #2, LIST_FA0	2184
			7E	52	A2 3C 000BE	MOVZBL 84(BUF), -(SP)	2188
			50	50	A2 3C 000C2	MOVZWL 82(BUF), -(SP)	
			51	55	A2 9A 000C6	MOVZWL 80(BUF), R0	
			51	10	78 000CA	MOVZBL 85(BUF), R1	
			51	6140	C6 9F 000CE	ASHL #16, R1, R1	
				00D5	C6 9F 000D1	PUSHAB (R1)[R0]	
			65	04	FB 000D5	PUSHAB P.AFE	
			64	00	FB 000D8	CALLS #4, LIST_FA0	2189
			7E	5A	A2 9A 000DB	CALLS #0, LIST_EOL	
			7E	58	A2 3C 000DF	MOVZBL 90(BUF), -(SP)	2193
			50	56	A2 3C 000E3	MOVZWL 88(BUF), -(SP)	
			51	58	A2 9A 000E7	MOVZWL 86(BUF), R0	
			51	10	78 000EB	MOVZBL 91(BUF), R1	
			51	6140	C6 9F 000EF	ASHL #16, R1, R1	
				00EF	C6 9F 000F2	PUSHAB (R1)[R0]	
			65	04	FB 000F6	PUSHAB P.AFF	
			64	00	FB 000F9	CALLS #4, LIST_FA0	2194
				5C	A2 9F 000FC	CALLS #0, LIST_EOL	
			65	0109	C6 9F 000FF	PUSHAB 92(BUF)	2195
			64	02	FB 00103	PUSHAB P.AFG	
				00DC	C2 DD 00109	CALLS #2, LIST_FA0	2196
			64	00	FB 00106	CALLS #0, LIST_EOL	2197
				00E0	C2 DD 0010D	PUSHL 220(BUF)	
			65	011D	C6 9F 00111	PUSHL 224(BUF)	
			64	03	FB 00115	PUSHAB P.AFH	
				00E4	C2 DD 0011B	CALLS #3, LIST_FA0	2198
			65	00	FB 00118	CALLS #0, LIST_EOL	2199
				0132	C6 9F 0011F	PUSHL 228(BUF)	
			64	02	FB 00123	PUSHAB P.AFI	
			64	00	FB 00126	CALLS #2, LIST_FA0	2200
			64	00	FB 00129	CALLS #0, LIST_EOL	2201
				FC85	C6 9F 0012C	PUSHAB ANALYZE_ONE_RECORD	2206
					53 DD 00130	PUSHL R3	
			00000000G	00	02 FB 00132	CALLS #2, DEBLOCK	
					04 00139	RET	2207

; Routine Size: 314 bytes. Routine Base: CODE + 0A9D

```
659 2208 1 %SBTTL 'ANALYZE - main analyze routine'  
660 2209 1 GLOBAL ROUTINE ANALYZE: NOVALUE=  
661 2210 1 !++  
662 2211 1  
663 2212 1  
664 2213 1 FUNCTIONAL DESCRIPTION:  
665 2214 1 This routine is the driver for analysis generation.  
666 2215 1  
667 2216 1 INPUT PARAMETERS:  
668 2217 1 NONE  
669 2218 1  
670 2219 1 IMPLICIT INPUTS:  
671 2220 1 NONE  
672 2221 1  
673 2222 1 OUTPUT PARAMETERS:  
674 2223 1 NONE  
675 2224 1  
676 2225 1 IMPLICIT OUTPUTS:  
677 2226 1 NONE  
678 2227 1  
679 2228 1 ROUTINE VALUE:  
680 2229 1 NONE  
681 2230 1  
682 2231 1 SIDE EFFECTS:  
683 2232 1 NONE  
684 2233 1  
685 2234 1 !--  
686 2235 1  
687 2236 2 BEGIN  
688 2237 2  
689 2238 2 LOCAL  
690 2239 2 BCB, ! Pointer to buffer control block  
691 2240 2 CHK_SAVESET: ! Check save set encryption on frst pass  
692 2241 2 PSAREA: ! Impure area  
693 2242 2  
694 2243 2 GLOBAL REGISTER  
695 2244 2 PS = 11: REF VECTOR; ! Impure area base register  
696 2245 2  
697 2246 2 BUILTIN  
698 2247 2 FP;  
699 2248 2  
700 2249 2  
701 2250 2 ! Establish the handler.  
702 2251 2  
703 2252 2 .FP = RESTORE_HANDLER;  
704 2253 2  
705 2254 2  
706 2255 2 ! Initialize impure area.  
707 2256 2  
708 2257 2 PS = PSAREA;  
709 2258 2 LIST_DESC[0] = LIST_SIZE;  
710 2259 2 LIST_DESC[1] = LIST_BUFFER;  
711 2260 2 CHK_SAVESET = 1;  
712 2261 2  
713 2262 2  
714 2263 2 ! Do the listing.  
715 2264 2
```

```

716 2265 2 INIT_IN_SAVE(FALSE);
717 2266 2 WHILE (BCB = READ_BUFFER()) NEQ 0 DO
718 2267 BEGIN
719 2268
720 2269 3 IF .QUAL[QUAL_SS_ENCRYP]
721 2270 3 THEN
722 2271 4 BEGIN
723 2272 4 | If we are decrypting an encrypted saveset and we don't yet have
724 2273 4 | the required decrypt context, do a special scan of the first block
725 2274 4 | to locate the backup summary record, extract the datakey information
726 2275 4 | and initialize the decryption context.
727 2276 4
728 2277 4 IF .CRYP_DATA_CODE EQLU 0
729 2278 5 THEN IF NOT (QUAL[QUAL_SS_ENCRYP] = CRYPTO_INIDECK(.BCB))
730 2279 4 | THEN SIGNAL(BACKUPS_BACNOTENC);
731 2280 4
732 2281 4 | Decrypt the buffer
733 2282 4
734 2283 4 CRYPTO_DECR_BLOCK(.BCB);
735 2284 4
736 2285 4 END
737 2286 3 ELSE
738 2287 4 BEGIN
739 2288 4 | Make sure that the save set is not encrypted.
740 2289 4
741 2290 4 IF .CHK_SAVESET
742 2291 4 THEN IF CRYPTO_CHKSAV(.BCB)
743 2292 4 | THEN SIGNAL(BACKUPS_ENCSAVSET) ;
744 2293 4 | CHK_SAVESET = 0 ;
745 2294 4 | END ;
746 2295 3
747 2296 3
748 2297 3 ANALYZE_ONE_BUFFER(.BCB);
749 2298 3
750 2299 2 END;
751 2300 2
752 2301 2 FIN_IN_SAVE(FALSE);
753 2302 1 END;

```

			083C 00000	.ENTRY	ANALYZE, Save R2,R3,R4,R5,R11	: 2209
	55 00000000G	00	9E 00002	MOVAB	LIB\$SIGNAL, R5	
	54 00000000	EF	9E 00009	MOVAB	QUAL+12, R4	
	5E FEF8	CE	9E 00010	MOVAB	-264(SP), SP	
	6D 00000000G	00	9E 00015	MOVAB	RESTORE_HANDLER, (FP)	
	5B	6E	9E 0001C	MOVAB	PSAREA, PS	
	68 0100	8F	3C 0001F	MOVZWL	#256 (PS)	
04	AB 08	AB	9E 00024	MOVAB	8(R11), 4(PS)	
	53	01	90 00029	MOVB	#1, (CHK_SAVESET	
		7E	D4 0002C	CLRL	- (SP)	
	00000000G 00	01	FB 0002E	CALLS	#1, INIT_IN_SAVE	
	00000000G 00	00	FB 00035	CALLS	#0, READ_BUFFER	
	52	50	D0 0003C	MOVL	R0, BCB	
		54	13 0003F	BEQL	6\$	
			18:			

ANALYZE
V04-000Analyze a save set
ANALYZE - main analyze routineL 4
15-Sep-1984 23:40:04 14-Sep-1984 11:53:45 VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]ANALYZE.B32;1Page 33
(6)

2C	02	A4	04	E1 00041	BBC	#4, QUAL+14, 3\$	2269
			06DC	C4 D5 00046	TSTL	CRYPTO_DATA_CODE	2277
02	A4	01	00000000G 00	1B 12 0004A	BNEQ	2\$	2278
				52 DD 0004C	PUSHL	BCB	
			04	01 FB 0004E	CALLS	#1, CRYPTO_INIDEC	
			09	50 F0 00055	INSV	RO, #4, #1, QUAL+14	
			65	50 E8 00058	BLBS	RO, 2\$	
			00000000G 00	8F DD 0005E	PUSHL	#BACKUPS_BACNOTENC	2279
			15	01 FB 00064	CALLS	#1, LIB\$SIGNAL	
			09	52 DD 00067	2\$:	PUSHL	2283
			65	01 FB 00069	CALLS	#1, CRYPTO_DECR_BLOCK	
			00000000G 00	1A 11 00070	BRB	5\$	2269
			15	53 E9 00072	3\$:	BLBC	2291
			09	52 DD 00075	PUSHL	CHK_SAVESET, 4\$	
			65	01 FB 00077	CALLS	#1, CRYPTO_CHKSAV	2292
			00000000G 00	50 E9 0007E	BLBC	RO, 4\$	
			15	8F DD 00081	PUSHL	#BACKUPS_ENCSAVSET	2293
			09	01 FB 00087	CALLS	#1, LIB\$SIGNAL	
			65	53 94 0008A	4\$:	CLR	2294
			FE33 CF	52 DD 0008C	5\$:	CHK_SAVESET	
			00000000G 00	01 FB 0008E	PUSHL	#1, ANALYZE_ONE_BUFFER	2297
				A0 11 00093	CALLS	1\$	2266
				7E D4 00095	6\$:	BRB	2301
				01 FB 00097	CLRL	-(SP)	
				04 0009E	CALLS	#1, FIN_IN_SAVE	2302
					RET		

: Routine Size: 159 bytes. Routine Base: CODE + 0BD7

ANALYZE
V04-000 Analyze a save set
ANALYZE - main analyze routine
:
: 755 2303 1 END
: 756 2304 0 ELUDOM

M 4
15-Sep-1984 23:40:04 VAX-11 Bli\$-32 v4.0-742
14-Sep-1984 11:53:45 [BACKUP.SRC]ANALYZE.B32;1

Page 34
(7)

.EXTRN LIB\$SIGNAL

PSECT SUMMARY

Name	Bytes	Attributes
COMMON	2124 NOVEC, WRT, RD ,NOEXE,NOSHR, LCL, REL, OVR,NOPIC,ALIGN(2)	
CODE	3190 NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)	

Library Statistics

File	-----	Symbols	-----	Pages	Processing
	Total	Loaded	Percent	Mapped	Time
\$_\$255\$DUA28:[SYSLIB]STARLET.L32;1	9776	9	0	581	00:01.2

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS\$:ANALYZE/OBJ=OBJ\$:ANALYZE MSRC\$:ANALYZE/UPDATE=(ENH\$:ANALYZE)
:
: Size: 1398 code + 3916 data bytes
: Run Time: 00:37.7
: Elapsed Time: 01:44.5
: Lines/CPU Min: 3662
: Lexemes/CPU-Min: 38872
: Memory Used: 382 pages
: Compilation Complete

0010 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

BACKUPMSG
LIS

ANALYZE
LIS

BUFFERS
LIS

CREATEDIR
LIS

BADBLOCK
LIS

BACKUPCMD
LIS

COMMAND
LIS